



EJU

REC'D 22 MAY 2000

WIPO

PCT

FR00/1137

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le **10 MAI 2000**

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE

26 bis, rue de Saint Petersburg
75800 PARIS Cédex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30

THIS PAGE BLANK (USPTO)

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

REQUÊTE EN DÉLIVRANCE

Confirmation d'un dépôt par télécopie ☐

Cet imprimé est à remplir à l'encre noire en lettres capitales

Réservé à l'INPI

DATE DE REMISE DES PIÈCES **29 AVR. 1999**
N° D'ENREGISTREMENT NATIONAL **99 05444**
DÉPARTEMENT DE DÉPÔT **75**
DATE DE DÉPÔT **29 AVR. 1999**

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE
À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

CABINET BEAU DE LOMENIE
158, rue de l'Université
75340 PARIS CEDEX 07

2 DEMANDE Nature du titre de propriété industrielle

☒ brevet d'invention

☐ demande divisionnaire

☐ certificat d'utilité

☐ transformation d'une demande
de brevet européen



demande initiale

☐ brevet d'invention

n° du pouvoir permanent

références du correspondant

téléphone

H25638/4/GYD

01.44.18.89.00

date

Établissement du rapport de recherche

☐ différé

☒ immédiat

Le demandeur, personne physique, requiert le paiement échelonné de la redevance

☐ oui

☐ non

Titre de l'invention (200 caractères maximum)

**"Procédé de gestion de commandes dans plusieurs fichiers d'application et
carte à puce pour la mise en oeuvre du procédé".**

3 DEMANDEUR (S) n° SIREN

code APE-NAF

Nom et prénoms (souligner le nom patronymique) ou dénomination

SCHLUMBERGER SYSTEMES

Forme juridique

SOCIETE ANONYME

Nationalité (s) **FRANCAISE**

Adresse (s) complète (s)

50, avenue Jean Jaurès
92120 Montrouge

Pays

FR

En cas d'insuffisance de place, poursuivre sur papier libre ☐

4 INVENTEUR (S) Les inventeurs sont les demandeurs

☐ oui

☒ non

Si la réponse est non, fournir une désignation séparée

5 RÉDUCTION DU TAUX DES REDEVANCES

☐ requise pour la 1ère fois

☐ requise antérieurement au dépôt ; joindre copie de la décision d'admission

6 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE

pays d'origine

numéro

date de dépôt

nature de la demande

7 DIVISIONS

antérieures à la présente demande

n°

date

n°

date

8 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE

(nom et qualité du signataire)

DRONNE GUY **92-3018**

SIGNATURE DU PRÉPOSÉ À LA RÉCEPTION

SIGNATURE APRÈS ENREGISTREMENT DE LA DEMANDE À L'INPI

DÉSIGNATION DE L'INVENTEUR

(si le demandeur n'est pas l'inventeur ou l'unique inventeur)

DEPARTEMENT DES BREVETS

26bis, rue de Saint-Petersbourg

75800 Paris Cédex 08

Tél. : 01 53 04 53 04 - Télécopie : 01 42 93 59 30

N° D'ENREGISTREMENT NATIONAL

0505444

TITRE DE L'INVENTION :

"Procédé de gestion de commandes dans plusieurs fichiers d'application et carte à puce pour la mise en oeuvre du procédé".

LE(S) SOUSSIGNÉ(S)

SCHLUMBERGER SYSTEMES

Société Anonyme

DÉSIGNE(NT) EN TANT QU'INVENTEUR(S) (indiquer nom, prénoms, adresse et souligner le nom patronymique) :

GRELLIER Stéphane J.

13, route de Vanne

45620 ISDES

FRANCE

NOTA : A titre exceptionnel, le nom de l'inventeur peut être suivi de celui de la société à laquelle il appartient (société d'appartenance) lorsque celle-ci est différente de la société déposante ou titulaire.

Date et signature (s) du (des) demandeur (s) ou du mandataire



PARIS LE 29 AVRIL 1999
CABINET BEAU DE LOMENIE
Mr DRONNE Guy
CPI N° 92-3018

L'invention concerne un procédé de gestion de commandes dans plusieurs fichiers d'application et une carte à puce adaptée à gérer des commandes dans plusieurs fichiers d'application.

On connaît des procédés de gestion de commandes adaptés à gérer des commandes dans plusieurs fichiers d'application. Ces procédés sont par exemple adaptés à gérer des commandes telles que créer ou ouvrir ou supprimer un sous-fichier dans un premier fichier concernant une première application ou telles que créer ou ouvrir ou supprimer un sous-fichier d'un deuxième fichier concernant une deuxième application. Les moyens mettant en oeuvre de tels procédés sont appelés « operating system » en anglais.

Un utilisateur qui souhaite ordonner l'exécution d'une telle opération doit satisfaire à la condition d'accès associée à la commande dans le fichier d'application.

Dans certains domaines d'application, ces conditions d'accès ont été normalisées. On connaît ainsi par exemple la norme EN 726-3 pour les procédés de gestion de commandes dans les cartes à puces. Cette norme mentionne par exemple une condition d'accès notée « ALW » pour laquelle l'utilisateur n'a besoin de satisfaire à aucune condition : ainsi une opération dans un fichier d'application, à laquelle est associée la condition d'accès « ALW » sera toujours exécutée si l'utilisateur l'ordonne. Cette norme mentionne aussi une condition d'accès notée « NEV » : si l'utilisateur ordonne l'exécution d'une opération dans un fichier d'application, à laquelle est associée la condition d'accès « NEV », cette opération ne sera pas exécutée. Cette norme mentionne également, entre autres, des conditions d'accès notées « CHV1 » et « CHV2 ».

Les procédés de gestion de commandes dans plusieurs fichiers d'application connus ne permettent pas de gérer conjointement des commandes pour des fichiers d'application différents. Ainsi, lorsqu'un opérateur fait exécuter des commandes dans un premier fichier d'application, qu'il passe à un deuxième fichier d'application pour y faire exécuter des commandes, et qu'il retourne au premier fichier d'application, il doit satisfaire à nouveau aux conditions d'accès pour faire

exécuter, dans le premier fichier d'application, des commandes analogues à celles qu'il a fait exécuter initialement.

L'objet de l'invention est de proposer un procédé de gestion de commandes dans plusieurs fichiers d'application, mis en œuvre aisément et en toute sécurité, ayant un coût réduit et permettant de résoudre ce problème.

Plus précisément, l'invention concerne un procédé de gestion de commandes dans plusieurs fichiers d'application, à chaque commande dans chaque fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à ladite commande est mémorisée, caractérisé en ce qu'on garde mémorisées les conditions d'accès conjointement pour tous les fichiers d'application.

De préférence, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application,

on recherche si la condition d'accès correspondant à ladite commande est mémorisée,

si la condition d'accès est mémorisée alors on lance l'exécution de la commande,

sinon on demande à l'utilisateur de satisfaire à la condition d'accès et alors, si l'utilisateur satisfait à la condition d'accès, on mémorise la condition d'accès et on lance l'exécution de la commande.

Avantageusement, n désignant un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions d'accès, on garde mémorisées les n dernières conditions d'accès satisfaites.

De préférence, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant pas en mémoire, n conditions d'accès étant mémorisées, on supprime de la mémoire la condition d'accès la plus anciennement satisfaite et on mémorise la condition d'accès nouvellement satisfaite.

Selon un premier mode de mise en œuvre du procédé, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

5 Selon un deuxième mode de mise en œuvre du procédé, n est supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

Selon un troisième mode de mise en œuvre du procédé, n est inférieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

10 Avantageusement, un contact étant établi pour démarrer ledit procédé de gestion de commandes, le contact étant rompu, on efface toutes les conditions d'accès.

L'invention concerne également une carte à puce adaptée à gérer des commandes dans plusieurs fichiers d'application, à chaque commande dans chaque
15 fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à ladite commande est en mémoire, caractérisée en ce qu'elle comporte des moyens pour garder en mémoire les conditions d'accès
20 conjointement pour tous les fichiers d'application.

De préférence, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application, elle comporte :

- des moyens adaptés à rechercher si la condition d'accès correspondant à ladite commande est en mémoire,
- 25 - des moyens adaptés à lancer l'exécution de la commande si la condition d'accès est en mémoire,
- des moyens adaptés à demander à l'utilisateur de satisfaire à la condition d'accès si la condition d'accès n'est pas en mémoire, et
- des moyens adaptés à mémoriser la condition d'accès et à lancer
30 l'exécution de la commande si l'utilisateur satisfait à la condition d'accès.

Avantageusement, n désignant un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions d'accès, elle comporte des moyens adaptés à garder en mémoire les n dernières conditions d'accès satisfaites.

De préférence, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant pas en mémoire, n conditions d'accès étant mémorisées, elle comporte des moyens adaptés à supprimer de la mémoire la condition d'accès la plus anciennement satisfaite et à mémoriser la condition d'accès nouvellement satisfaite.

Selon un premier mode de réalisation, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

Selon un deuxième mode de réalisation, n est supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

Selon un troisième mode de réalisation, n est inférieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

Avantageusement, la carte comporte un contact et des moyens adaptés à supprimer toutes les conditions d'accès lorsque le contact est rompu.

D'autres caractéristiques et avantages de l'invention apparaîtront au cours de la description d'un mode de réalisation qui va suivre, donné à titre d'exemple non limitatif, à l'appui des cinq figures sur lesquelles :

la figure 1 représente un tableau de mémorisation relatif à un espace mémoire dans la mémoire vive de la carte à puce adaptée à gérer des commandes dans plusieurs fichiers d'application, le tableau est représenté dans l'état initial de mémorisation,

les figures 2 à 5 sont analogues à la figure 1 pour différents états de mémorisation,

la figure 6 représente la carte et son environnement de fonctionnement,

la figure 7 représente l'organisation logique de la carte, et

la figure 8 représente un organigramme logique des étapes du procédé de l'invention.

En se référant plus particulièrement à la figure 7, on va décrire l'organisation logique d'une carte 8 à puce adaptée à gérer des commandes dans plusieurs fichiers d'application.

La carte 8 comprend des moyens pour mettre en œuvre le procédé de gestion
5 de commandes, ce moyen est dénommé « OS » et porte la référence 3.

La carte 8 comprend deux fichiers d'application, aussi appelés répertoires.

Le premier fichier d'application est dénommé « compte bancaire » et porte la référence 40. Ce fichier d'application est prévu pour permettre à un utilisateur de gérer des données relatives à son compte bancaire.

10 Dans le fichier « compte bancaire », l'utilisateur peut ordonner l'exécution d'une commande dénommée « solde » portant la référence 41, d'une commande dénommée « débit » portant la référence 42 et d'une commande dénommée « transfert vers compte bourse » portant la référence 43.

La commande « solde » contient des instructions permettant à l'utilisateur de
15 connaître le solde de son compte bancaire.

La commande « débit » contient des instructions permettant à l'utilisateur d'ordonner un mouvement de débit de son compte bancaire.

La commande « transfert vers compte bourse » permet à l'utilisateur d'ordonner le transfert d'un certain montant du compte bancaire vers un compte
20 bourse.

Le deuxième fichier d'application est dénommé « bourse » et porte la référence 50. Ce fichier d'application est prévu pour permettre à l'utilisateur de gérer des données relatives à son portefeuille de bourse.

Dans le fichier « bourse », l'utilisateur peut ordonner l'exécution d'une
25 commande dénommée « solde compte bourse » portant la référence 51, d'une commande dénommée « solde bourse » portant la référence 52, d'une commande dénommée « titre » portant la référence 53, d'une commande dénommée « ordre d'achat » portant la référence 54, d'une commande dénommée « ordre de vente » portant la référence 55 et d'une commande dénommée « transfert vers compte
30 bancaire » portant la référence 56.

La commande « solde compte bourse » contient des instructions permettant à l'utilisateur de connaître le solde du compte bourse prévu pour alimenter et recevoir le montant correspondant aux achats et ventes de titres de bourse.

La commande « solde bourse » contient des instructions permettant à l'utilisateur de connaître la valeur globale des titres en bourse qu'il possède.

La commande « titre » contient des instructions permettant à l'utilisateur de connaître les titres de bourse que l'utilisateur possède.

La commande « ordre d'achat » permet à l'utilisateur de donner l'instruction d'acheter des titres de bourse.

La commande « ordre de vente » permet à l'utilisateur de donner l'instruction de vendre des titres de bourse.

La commande « transfert vers compte bancaire » permet à l'utilisateur de donner l'instruction de transférer un certain montant du compte bourse vers le compte bancaire.

Pour pouvoir effectuer les commandes « solde », « solde compte bourse », « solde bourse » ou « titre », l'utilisateur devra au préalable entrer des conditions d'accès notées « CHV1 ».

Pour pouvoir effectuer les commandes « débit », « transfert vers compte bourse », « ordre d'achat », « ordre de vente » ou « transfert vers compte bancaire », l'utilisateur devra au préalable entrer les conditions d'accès notées « CHV2 ».

La carte 8 comporte des moyens représentés par un tableau, pour garder en mémoire les conditions d'accès conjointement pour tous les fichiers d'application.

Le tableau comporte deux colonnes et quatre lignes. Le nombre de lignes est le nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application : la carte comprend en effet deux fichiers d'application et chacun d'eux comporte deux conditions d'accès pour l'ensemble de ses commandes.

Lorsqu'une condition d'accès dans un fichier d'application est mémorisée, l'adresse du fichier d'application figure sur une ligne de la première colonne et ladite condition d'accès figure sur la même ligne de la deuxième colonne. A l'inverse, lorsqu'une condition d'accès dans un fichier d'application n'est pas

mémorisée, l'adresse du fichier d'application ne figure sur aucune ligne de la première colonne.

En se référant plus particulièrement à la figure 6, on a représenté la carte 8 adaptée à être insérée dans un lecteur 12. Le lecteur 12 est connecté à un ordinateur 10 relié à un clavier 14, à un écran 16 et à un distributeur 18 de billets. L'ordinateur 10 est connecté à un ordinateur central 20 adapté à mémoriser les opérations qui sont demandées.

En se référant maintenant plus particulièrement à la figure 8, on a représenté les étapes du procédé de gestion de commandes dans plusieurs fichiers d'application.

Dans la situation initiale notée 80, l'utilisateur a ordonné l'exécution d'une commande dans un fichier d'application.

Au cours d'une première étape notée 81, on recherche si la condition d'accès correspondant à ladite commande est mémorisée.

Si la condition d'accès est mémorisée, alors on lance, au cours d'une étape notée 82, l'exécution de la commande.

Si la condition d'accès n'est pas mémorisée, alors, au cours d'une étape notée 83, on demande à l'utilisateur de satisfaire à la condition d'accès.

Et alors, si l'utilisateur satisfait à la condition d'accès : au cours d'une étape notée 84, on mémorise la condition d'accès et on lance l'exécution de la commande.

Un exemple d'utilisation de la carte va maintenant être décrit.

L'utilisateur insère la carte 8 dans le lecteur 12. A ce stade de l'utilisation, le tableau est initialisé par l'« OS » : le tableau est donc vide de toute donnée (figure 1).

L'utilisateur souhaite connaître le solde de son compte bancaire. Pour cela, l'utilisateur ordonne l'exécution de la commande « solde » dans le fichier « compte bancaire ».

L'« OS » recherche si la condition d'accès correspondant à la commande « solde » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 » du fichier d'application « compte bancaire » est mémorisée dans le tableau ; comme

ce n'est pas le cas, car le tableau est vide, l'« OS » demande à l'utilisateur de satisfaire à la condition d'accès « CHV1 ». L'utilisateur entre un code confidentiel sur le clavier et l'utilisateur satisfait ainsi à la condition d'accès « CHV1 ». L'« OS » mémorise la condition d'accès dans le tableau (figure 2) et lance
 5 l'exécution de la commande « solde » : le solde du compte bancaire de l'utilisateur apparaît à l'écran.

L'utilisateur souhaite ensuite effectuer un débit de son compte bancaire. Pour cela, l'utilisateur ordonne l'exécution de la commande « débit » dans le fichier « compte bancaire ».

10 L'« OS » recherche si la condition d'accès correspondant à la commande « débit » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV2 » du fichier d'application « compte bancaire » est mémorisée dans le tableau ; comme ce n'est pas le cas (figure 2), l'« OS » demande à l'utilisateur de satisfaire à la condition d'accès « CHV2 ». L'utilisateur entre un code confidentiel sur le clavier
 15 et l'utilisateur satisfait ainsi à la condition d'accès « CHV2 ». L'« OS » mémorise la condition d'accès dans le tableau (figure 3) et lance l'exécution de la commande « débit ». Le distributeur fournit à l'utilisateur la somme souhaitée.

L'utilisateur souhaite alors connaître le solde de son compte bancaire et vérifier que son compte bancaire a bien été mémorisé. Pour cela, l'utilisateur
 20 ordonne l'exécution de la commande « solde » dans le fichier « compte bancaire ».

L'« OS » recherche si la condition d'accès correspondant à la commande « solde » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 » du fichier d'application « compte bancaire » est mémorisée dans le tableau ; comme c'est le cas (figure 3), l'« OS » lance l'exécution de la commande « solde ». Le
 25 solde du compte bancaire de l'utilisateur apparaît à l'écran et indique bien le nouveau solde, c'est-à-dire, le montant indiqué lors de la première commande de l'opération « solde » diminué de la somme débitée lors de l'opération « débit ».

L'utilisateur souhaite connaître le solde de son compte bourse. Pour cela, l'utilisateur ordonne l'exécution de la commande « solde compte bourse » dans le
 30 fichier « bourse ».

L'« OS » recherche si la condition d'accès correspondant à la commande « solde compte bourse » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 » du fichier d'application « bourse » est mémorisée dans le tableau ; comme ce n'est pas le cas (figure 3), l'« OS » demande à l'utilisateur de
 5 satisfaire à la condition d'accès « CHV1 ». L'utilisateur entre un code confidentiel sur le clavier et l'utilisateur satisfait ainsi à la condition d'accès « CHV1 ». L'« OS » mémorise la condition d'accès dans le tableau (figure 4) et lance l'exécution de la commande « solde compte bourse » : le solde du compte bourse de l'utilisateur apparaît à l'écran.

10 L'utilisateur souhaite maintenant connaître les titres de bourse qu'il possède dans son portefeuille. Pour cela, l'utilisateur ordonne l'exécution de la commande « titre » dans le fichier « bourse ».

L'« OS » recherche si la condition d'accès correspondant à la commande « titre » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 »
 15 du fichier d'application « bourse » est mémorisée dans le tableau ; comme c'est le cas (figure 4), l'« OS » lance l'exécution de la commande « titre » : les titres de bourse possédés par l'utilisateur apparaissent à l'écran.

L'utilisateur souhaite maintenant vendre certains titres de bourse. Pour cela, l'utilisateur ordonne l'exécution de la commande « ordre de vente » dans le fichier
 20 « bourse ».

L'« OS » recherche si la condition d'accès correspondant à la commande « solde compte bourse » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV2 » du fichier d'application « bourse » est mémorisée dans le tableau ; comme ce n'est pas le cas (figure 4), l'« OS » demande à l'utilisateur de
 25 satisfaire à la condition d'accès « CHV2 ». L'utilisateur entre un code confidentiel sur le clavier et l'utilisateur satisfait ainsi à la condition d'accès « CHV2 ». L'« OS » mémorise la condition d'accès dans le tableau (figure 5) et lance l'exécution de la commande « ordre de vente » : l'« OS » passe ainsi l'ordre des titres que l'utilisateur vend.

30 L'utilisateur souhaite alors connaître le solde de son compte bourse et vérifier que son compte bourse a bien été crédité du montant des ventes des titres

effectuées. Pour cela, l'utilisateur ordonne l'exécution de la commande « solde compte bourse » dans le fichier « bourse ».

5 L'« OS » recherche si la condition d'accès correspondant à la commande « solde compte bourse » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 » du fichier d'application « bourse » est mémorisée dans le tableau ; comme c'est le cas (figure 5), l'« OS » lance l'exécution de la commande « solde compte bourse ». Le solde du compte bourse de l'utilisateur apparaît à l'écran et indique bien le nouveau solde, c'est-à-dire, le montant indiqué lors de la première commande de l'opération « solde compte bourse » augmenté de la
10 somme dégagée lors de la vente des titres de bourse.

L'utilisateur souhaite transférer une somme de son compte bourse sur son compte bancaire. Pour cela, l'utilisateur ordonne l'exécution de la commande « transfert vers compte bancaire » dans le fichier « bourse ».

15 L'« OS » recherche si la condition d'accès correspondant à la commande « transfert vers compte bancaire » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV2 » du fichier d'application « bourse » est mémorisée dans le tableau ; comme c'est le cas (figure 5), l'« OS » lance l'exécution de la commande « transfert vers compte bancaire ».

20 L'utilisateur souhaite alors connaître le solde de son compte bancaire et vérifier que son compte bancaire a bien été crédité. Pour cela, l'utilisateur ordonne l'exécution de la commande « solde » dans le fichier « compte bancaire ».

25 L'« OS » recherche si la condition d'accès correspondant à la commande « solde » est mémorisée, dans notre cas l'« OS » cherche si la condition « CHV1 » du fichier d'application « compte bancaire » est mémorisée dans le tableau ; comme c'est le cas (figure 5), l'« OS » lance l'exécution de la commande « crédit ». Le solde du compte bancaire de l'utilisateur apparaît à l'écran et indique bien le nouveau solde, c'est-à-dire, le montant indiqué lors de la deuxième commande de l'opération « solde » augmenté de la somme transférée lors de l'opération « transfert vers compte bancaire ».

L'utilisateur souhaite alors connaître le solde de son compte bourse et vérifier que son compte bourse a bien été débité. Pour cela, l'utilisateur ordonne l'exécution de la commande « solde compte bourse » dans le fichier « bourse ».

5 L'« OS » recherche si la condition d'accès correspondant à la commande « solde compte bourse » est mémorisée, dans le cas considéré l'« OS » cherche si la condition « CHV1 » du fichier d'application « bourse » est mémorisée dans le tableau ; comme c'est le cas (figure 5), l'« OS » lance l'exécution de la commande « solde compte bourse ». Le solde du compte bourse de l'utilisateur apparaît à l'écran et indique bien le nouveau solde, c'est-à-dire, le montant indiqué lors de la
10 première commande de l'opération « solde compte bourse » diminué de la somme transférée lors de l'opération « transfert vers compte bancaire ».

L'utilisateur décide de finir d'ordonner des commandes et ferme la session carte qu'il vient d'effectuer, ce faisant le tableau va être initialisé, c'est-à-dire que toutes les données vont être effacées. Puis l'utilisateur retire la carte du lecteur.

15 Pour des raisons de sécurité, si en cours d'utilisation de la carte, cette dernière est retirée du lecteur, les données du tableau sont automatiquement effacées.

Bien entendu, la présente invention n'est nullement limitée au mode d'exécution décrit et représenté qui n'a été donné qu'à titre d'exemple. En
20 particulier, elle comprend tous les moyens constituant des équivalents techniques des moyens décrits ainsi que leurs combinaisons, si celles-ci sont exécutées selon l'esprit de l'invention.

REVENDEICATIONS

1. Procédé de gestion de commandes dans plusieurs fichiers d'application, à chaque commande dans chaque fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à ladite commande est mémorisée, caractérisé en ce qu'on garde mémorisées les conditions d'accès conjointement pour tous les fichiers d'application.
2. Procédé selon la revendication 1, caractérisé en ce que, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application, on recherche si la condition d'accès correspondant à ladite commande est mémorisée, si la condition d'accès est mémorisée alors on lance l'exécution de la commande, sinon on demande à l'utilisateur de satisfaire à la condition d'accès et alors, si l'utilisateur satisfait à la condition d'accès, on mémorise la condition d'accès et on lance l'exécution de la commande.
3. Procédé selon la revendication 2, caractérisé en ce que, n désignant un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions d'accès, on garde mémorisées les n dernières conditions d'accès satisfaites.
4. Procédé selon la revendication 3, caractérisé en ce que, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant pas en mémoire, n conditions d'accès étant mémorisées, on supprime de la mémoire la condition d'accès la plus anciennement satisfaite et on mémorise la condition d'accès nouvellement satisfaite.
5. Procédé selon l'une quelconque des revendications 3 ou 4, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

6. Procédé selon l'une quelconque des revendications 3 ou 4, n est supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

7. Procédé selon l'une quelconque des revendications 3 ou 4, n est inférieur
5 au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

8. Procédé selon l'une quelconque des revendications 1 à 7, un contact étant établi pour démarrer ledit procédé de gestion de commandes, le contact étant rompu, on efface toutes les conditions d'accès.

9. Carte (8) à puce adaptée à gérer des commandes dans plusieurs fichiers
10 d'application, à chaque commande dans chaque fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à ladite commande est
15 en mémoire, caractérisée en ce qu'elle comporte des moyens pour garder en mémoire les conditions d'accès conjointement pour tous les fichiers d'application.

10. Carte (8) selon la revendication 9, caractérisée en ce que, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application, elle comporte :

- 20 - des moyens adaptés à rechercher si la condition d'accès correspondant à ladite commande est en mémoire,
- des moyens adaptés à lancer l'exécution de la commande si la condition d'accès est en mémoire,
- des moyens adaptés à demander à l'utilisateur de satisfaire à la condition
25 d'accès si la condition d'accès n'est pas en mémoire, et
- des moyens adaptés à mémoriser la condition d'accès et à lancer l'exécution de la commande si l'utilisateur satisfait à la condition d'accès.

11. Carte (8) selon la revendication 10, caractérisée en ce que, n désignant un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions
30 d'accès, elle comporte des moyens adaptés à garder en mémoire les n dernières conditions d'accès satisfaites.

12. Carte (8) selon la revendication 11, caractérisée en ce que, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant pas en mémoire, n conditions d'accès étant mémorisées, elle comporte des moyens adaptés à supprimer de la mémoire la condition d'accès la plus anciennement
5 satisfaite et à mémoriser la condition d'accès nouvellement satisfaite.

13. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

14. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est
10 supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

15. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est inférieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

16. Carte (8) selon l'une quelconque des revendications précédentes
15 caractérisée en ce qu'elle comporte un contact et des moyens adaptés à supprimer toutes les conditions d'accès lorsque le contact est rompu.

FIG.1

« Compte bancaire »	« CHV1 »

FIG.2

« Compte bancaire »	« CHV2 »
« Compte bancaire »	« CHV1 »

FIG.3

« Bourse »	« CHV1 »
« Compte bancaire »	« CHV2 »
« Compte bancaire »	« CHV1 »

FIG.4

« Bourse »	« CHV2 »
« Bourse »	« CHV1 »
« Compte bancaire »	« CHV2 »
« Compte bancaire »	« CHV1 »

FIG.5

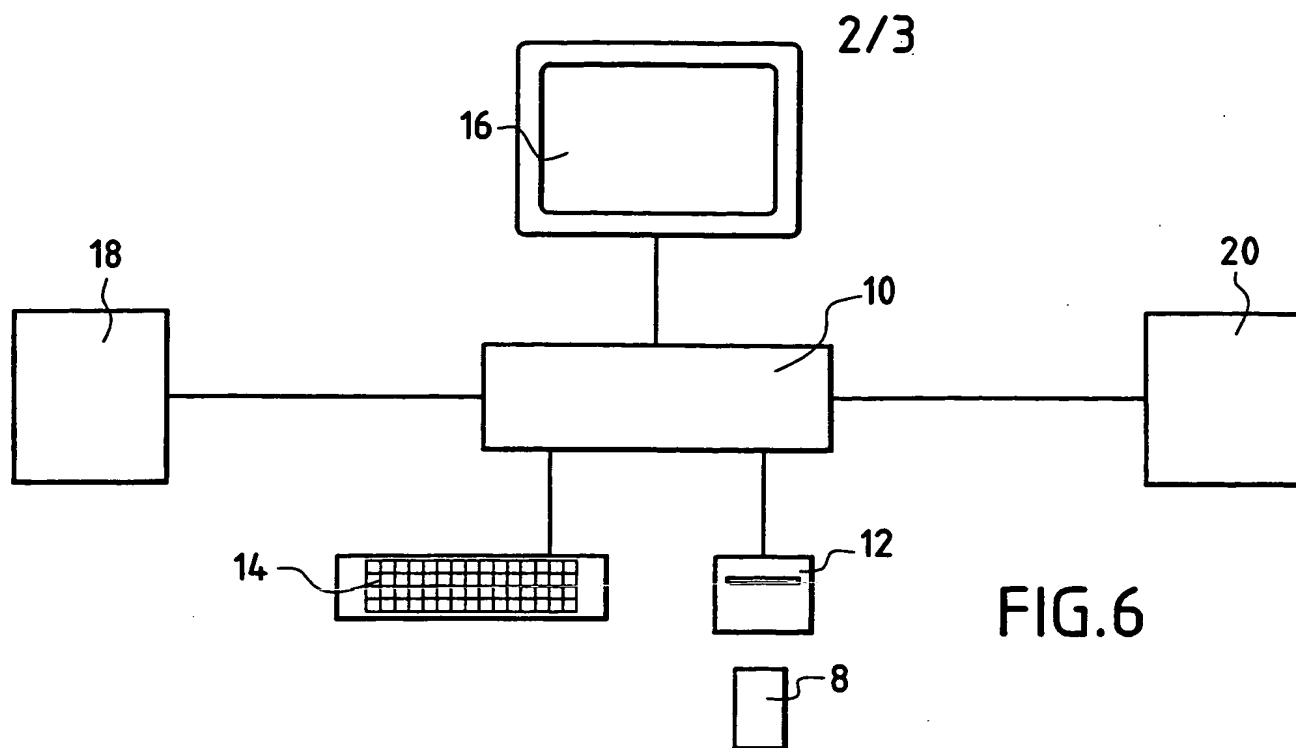


FIG.6

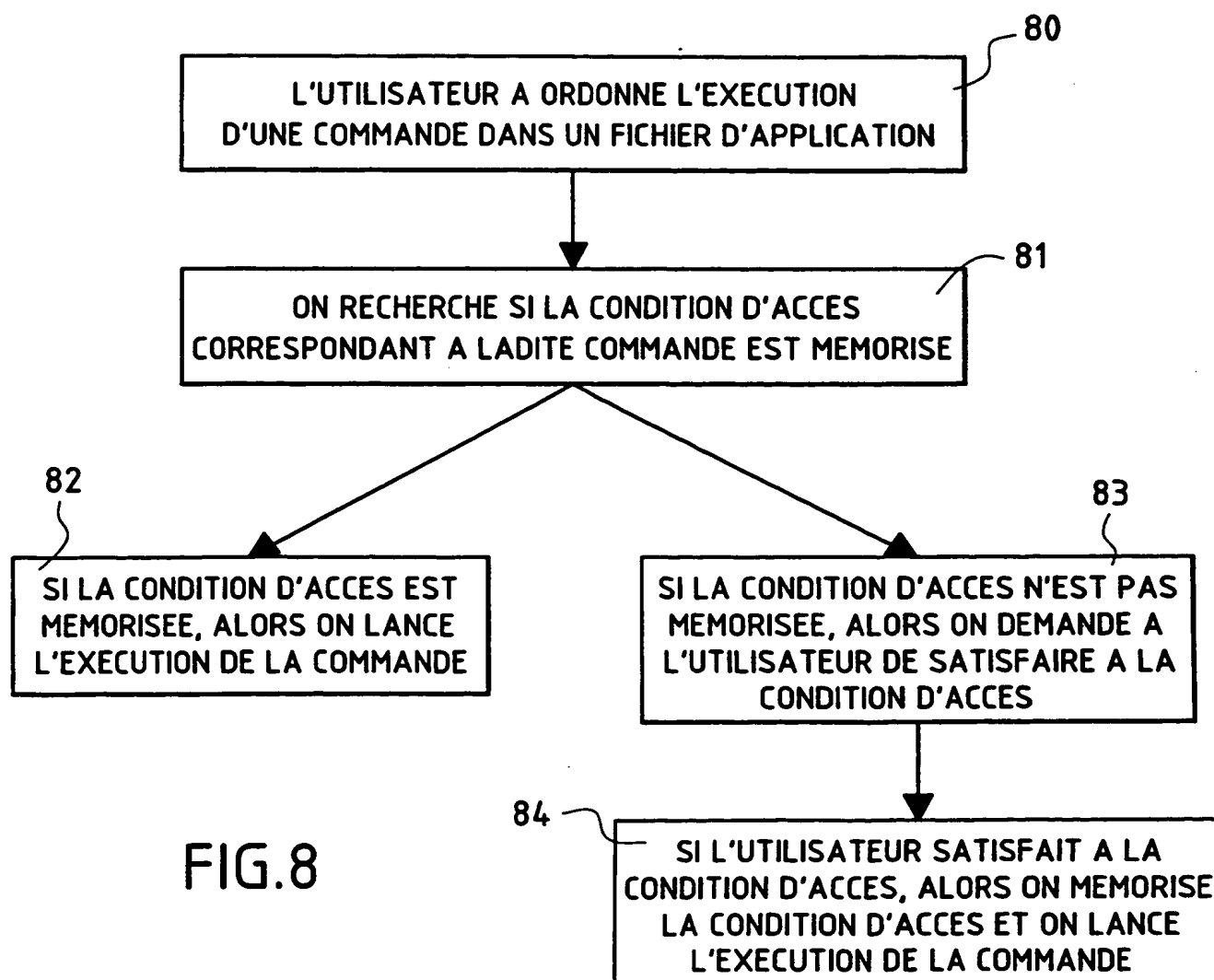


FIG.8

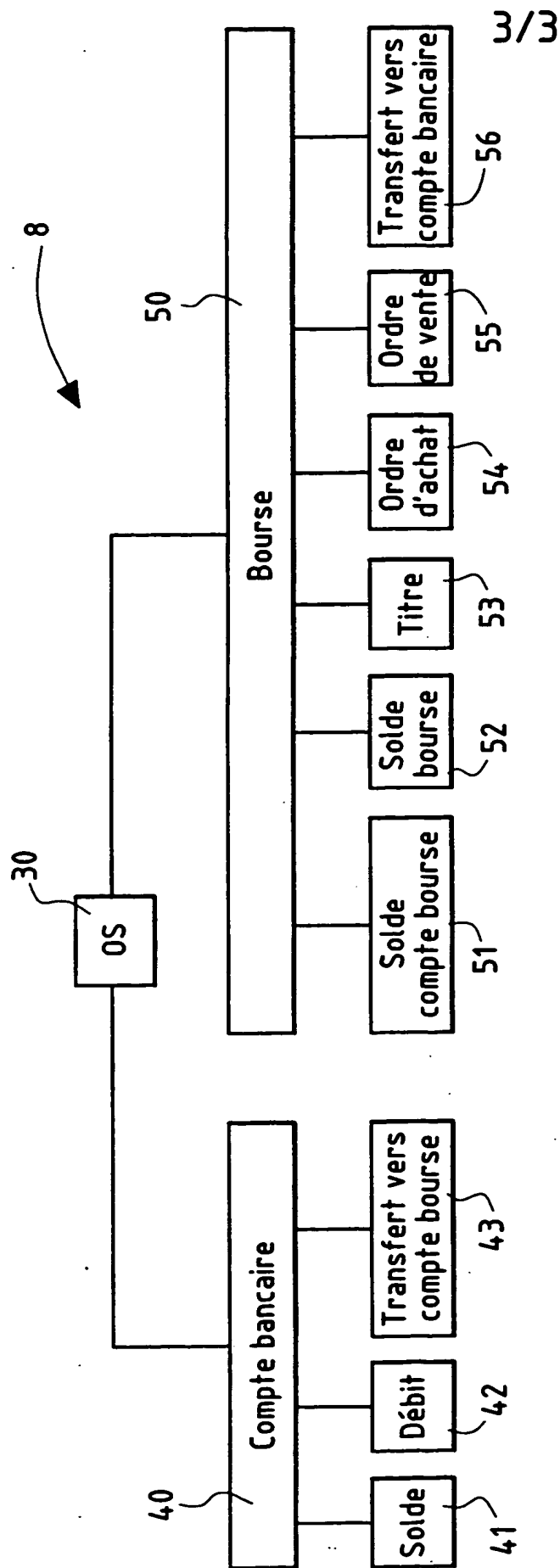


FIG.7

REVENDEICATIONS

1. Procédé de gestion de commandes dans plusieurs fichiers d'application, à chaque commande dans chaque fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans
5 chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à ladite commande est mémorisée, caractérisé en ce qu'on garde mémorisées les conditions d'accès conjointement pour tous les fichiers d'application auxquels l'utilisateur a déjà eu
10 accès.

2. Procédé selon la revendication 1, caractérisé en ce que, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application,

on recherche si la condition d'accès correspondant à ladite commande est mémorisée,

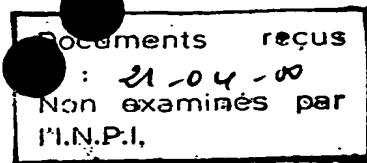
15 si la condition d'accès est mémorisée alors on lance l'exécution de la commande,

sinon on demande à l'utilisateur de satisfaire à la condition d'accès et alors, si l'utilisateur satisfait à la condition d'accès, on mémorise la condition d'accès et on lance l'exécution de la commande.

20 3. Procédé selon la revendication 2, caractérisé en ce que, n désignant un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions d'accès, on garde mémorisées les n dernières conditions d'accès satisfaites.

4. Procédé selon la revendication 3, caractérisé en ce que, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant
25 pas en mémoire, n conditions d'accès étant mémorisées, on supprime de la mémoire la condition d'accès la plus anciennement satisfaite et on mémorise la condition d'accès nouvellement satisfaite.

5. Procédé selon l'une quelconque des revendications 3 ou 4, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers
30 d'application.



6. Procédé selon l'une quelconque des revendications 3 ou 4, n est supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

5 7. Procédé selon l'une quelconque des revendications 3 ou 4, n est inférieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

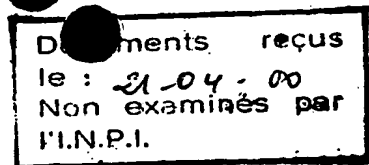
8. Procédé selon l'une quelconque des revendications 1 à 7, un contact étant établi pour démarrer ledit procédé de gestion de commandes, le contact étant rompu, on efface toutes les conditions d'accès.

10 9. Carte (8) à puce adaptée à gérer des commandes dans plusieurs fichiers d'application, à chaque commande dans chaque fichier d'application étant associée une condition d'accès correspondant à ladite commande, chaque commande dans chaque fichier d'application étant exécutée si un utilisateur a ordonné l'exécution de ladite commande et si la condition d'accès correspondant à
15 ladite commande est en mémoire, caractérisée en ce qu'elle comporte des moyens pour garder en mémoire les conditions d'accès conjointement pour tous les fichiers d'application auxquels l'utilisateur a déjà eu accès.

10. Carte (8) selon la revendication 9, caractérisée en ce que, un utilisateur ayant ordonné l'exécution d'une commande dans un fichier d'application, elle
20 comporte :

- des moyens adaptés à rechercher si la condition d'accès correspondant à ladite commande est en mémoire,
- des moyens adaptés à lancer l'exécution de la commande si la condition d'accès est en mémoire,
- 25 - des moyens adaptés à demander à l'utilisateur de satisfaire à la condition d'accès si la condition d'accès n'est pas en mémoire, et
- des moyens adaptés à mémoriser la condition d'accès et à lancer l'exécution de la commande si l'utilisateur satisfait à la condition d'accès.

11. Carte (8) selon la revendication 10, caractérisée en ce que, n désignant
30 un entier supérieur ou égal à deux, l'utilisateur ayant satisfait à n conditions



d'accès, elle comporte des moyens adaptés à garder en mémoire les n dernières conditions d'accès satisfaites.

12. Carte (8) selon la revendication 11, caractérisée en ce que, une nouvelle condition d'accès étant satisfaite par l'utilisateur, ladite condition d'accès n'étant pas en mémoire, n conditions d'accès étant mémorisées, elle comporte des moyens adaptés à supprimer de la mémoire la condition d'accès la plus anciennement satisfaite et à mémoriser la condition d'accès nouvellement satisfaite.

13. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est égal au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

14. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est supérieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

15. Carte (8) selon l'une quelconque des revendications 11 ou 12, n est inférieur au nombre de conditions d'accès pour l'ensemble des commandes de tous les fichiers d'application.

16. Carte (8) selon l'une quelconque des revendications précédentes caractérisée en ce qu'elle comporte un contact et des moyens adaptés à supprimer toutes les conditions d'accès lorsque le contact est rompu.